ANTLAB
ADVANCED NETWORK TECHNOLOGIES LAB

POLITECNICO MILANO 1863

Developing EFSM-based stateful applications with FlowBlaze.p4 and ONOS

Daniele Moro, Davide Sanvito^, Antonio Capone**

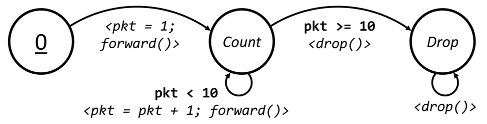
* Politecnico di Milano, Italy
^ NEC Laboratories Europe, Germany

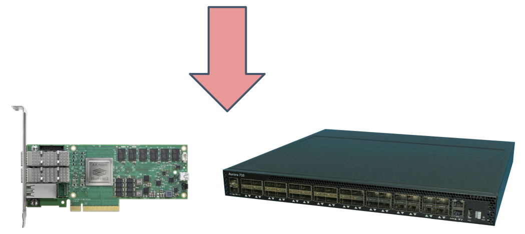
3rd P4 Workshop in Europe (EuroP4)
December 1, 2020
Barcelona (Spain) - Online Workshop

Introduction

5G and Mobile Edge Computing requires offloading of network functions to data plane

- **P4**: reference language for data plane programming
- **State Machines**: powerful abstraction to develop stateful packet processing
- **FlowBlaze [NSDI ‘19]**: EFSM-based stateful packet processing architecture





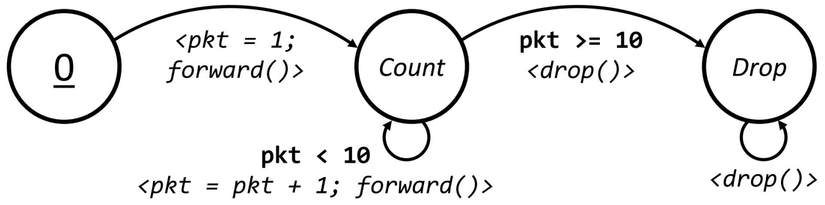
FlowBlaze currently lacks of:

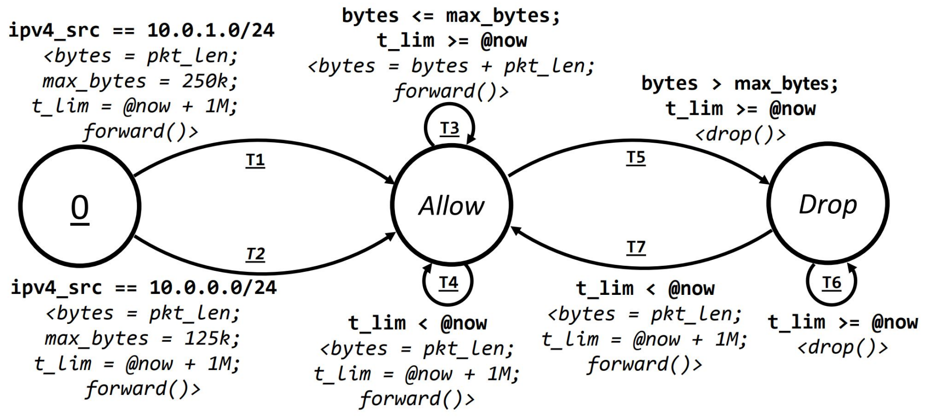
- Prototyping platform and P4 implementation*
- Integration with DC-style fabric

*FlowBlaze.p4 [NFV-SDN ‘20]

FlowBlaze.p4*

- FlowBlaze library implementation in P4
- Open source library
- GUI to automatically translate EFSM into table entries
- Exploit all the tools from the P4 Community
- Targets: BMv2 and V1Model





* D. Moro, et al. "***FlowBlaze.p4: a library for quick prototyping of stateful SDN applications in P4***" IEEE NFV-SDN 2020

FlowBlaze.p4 - configuration

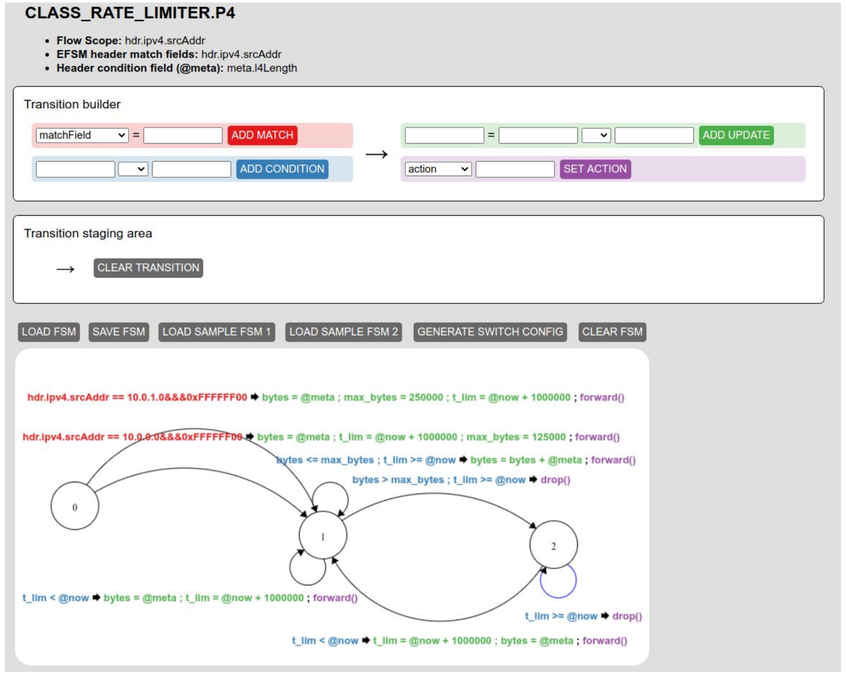
Compile-time configuration

VS

```
#include "../flowblaze_lib/flowblaze_metadata.p4"
#include "headers.p4"
#include "metadata.p4"
#include "../flowblaze_lib/flowblaze.p4"
 • • •
#define FLOW_SCOPE { hdr.ipv4.srcAddr }
#define CUSTOM_ACTIONS_DEFINITION @name(".FlowBlaze.forward") \
                                  action forward() { \
                                    \
                                  } \
                                  @name(".FlowBlaze.drop") \
                                  action drop() { \
                                    mark_to_drop(standard_metadata); \
                                    exit; \
                                  }
#define CUSTOM_ACTIONS_DECLARATION forward; drop;
// Configuration parameter left black because not needed
//     #define METADATA_OPERATION_COND
//     #define EFSM_MATCH_FIELDS
//     #define CONTEXT_TABLE_SIZE
 • • •
 apply {
     if (hdr.ethernet.isValid()) {
         FlowBlaze.apply(hdr, meta, standard_metadata);
         t_l2_fwd.apply();
     }
 }
```

Run-time configuration

- Draw the EFSM
- Auto generate the configuration via Python backend

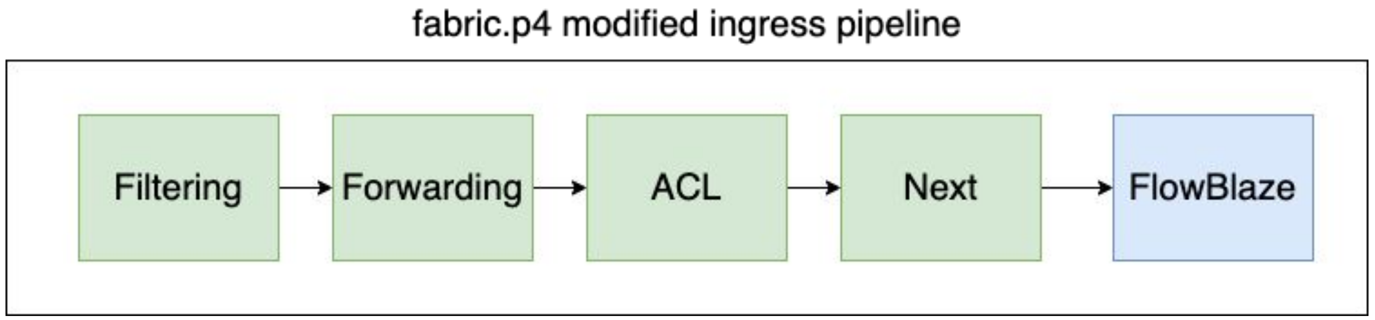

CLASS_RATE_LIMITER.P4

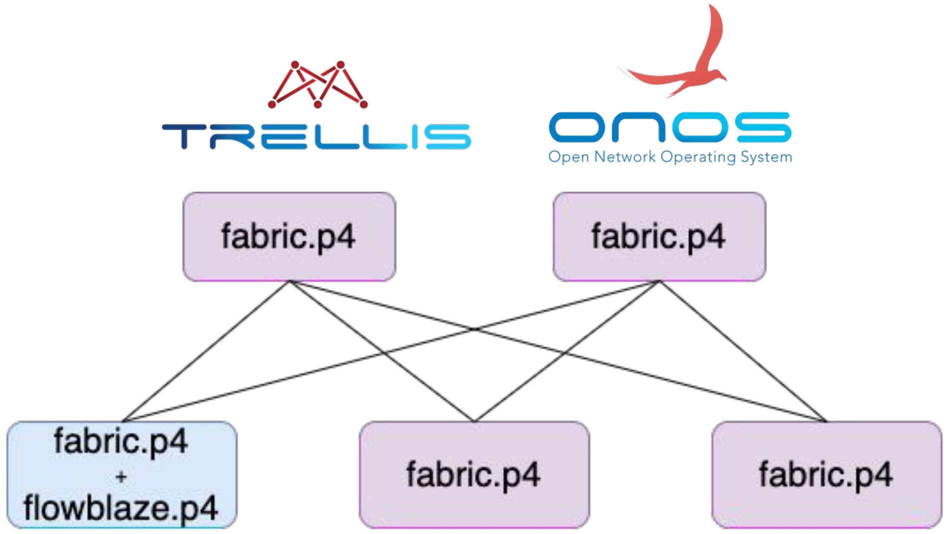
- **Flow Scope:** hdr.ipv4.srcAddr
- **EFSM header match fields:** hdr.ipv4.srcAddr
- **Header condition field (@meta):** meta.l4Length

```
hdr.ipv4.srcAddr == 10.0.1.0&&&0xFFFFFF00 ➡ bytes = @meta ; max_bytes = 250000 ; t_lim = @now + 1000000 ; forward()
hdr.ipv4.srcAddr == 10.0.0.0&&&0xFFFFFF00 ➡ bytes = @meta ; t_lim = @now + 1000000 ; max_bytes = 125000 ; forward()
bytes <= max_bytes ; t_lim >= @now ➡ bytes = bytes + @meta ; forward()
bytes > max_bytes ; t_lim >= @now ➡ drop()
t_lim < @now ➡ bytes = @meta ; t_lim = @now + 1000000 ; forward()
t_lim >= @now ➡ drop()
t_lim < @now ➡ t_lim = @now + 1000000 ; bytes = @meta ; forward()
```

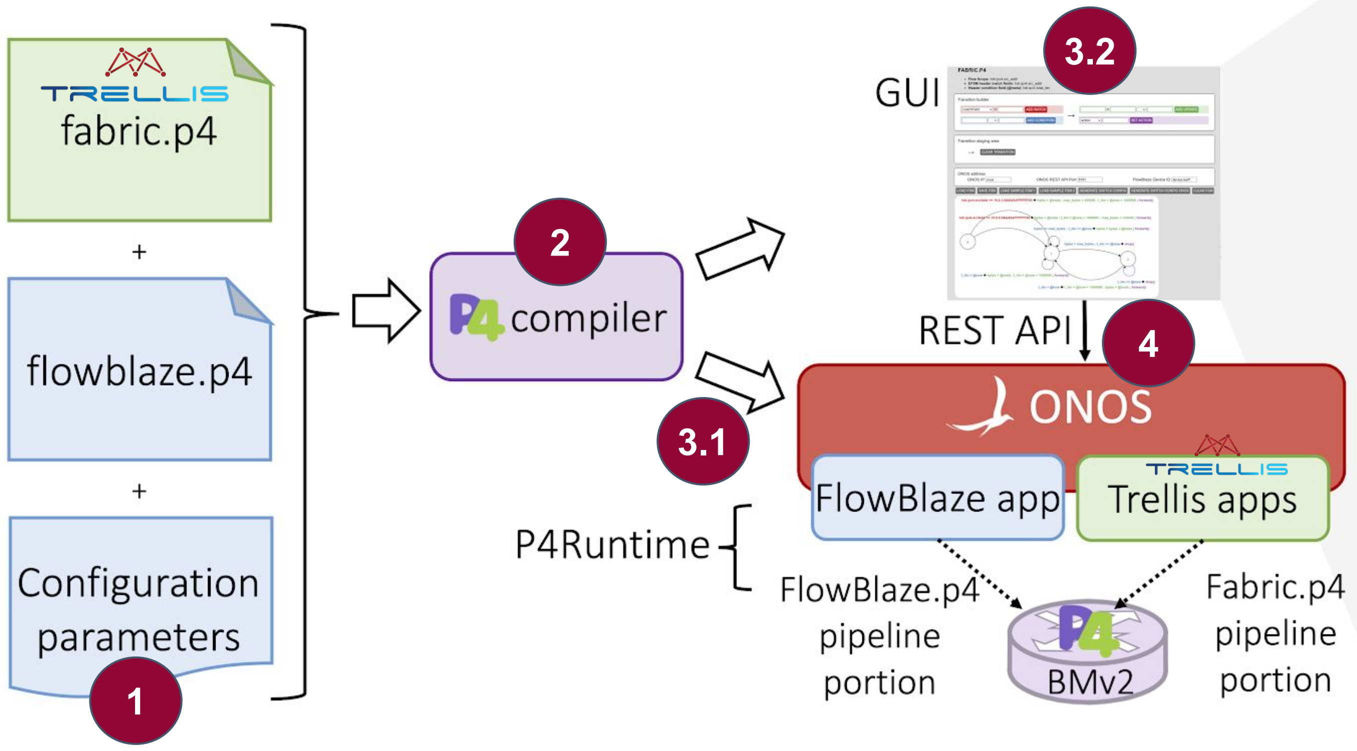
ONOS integration

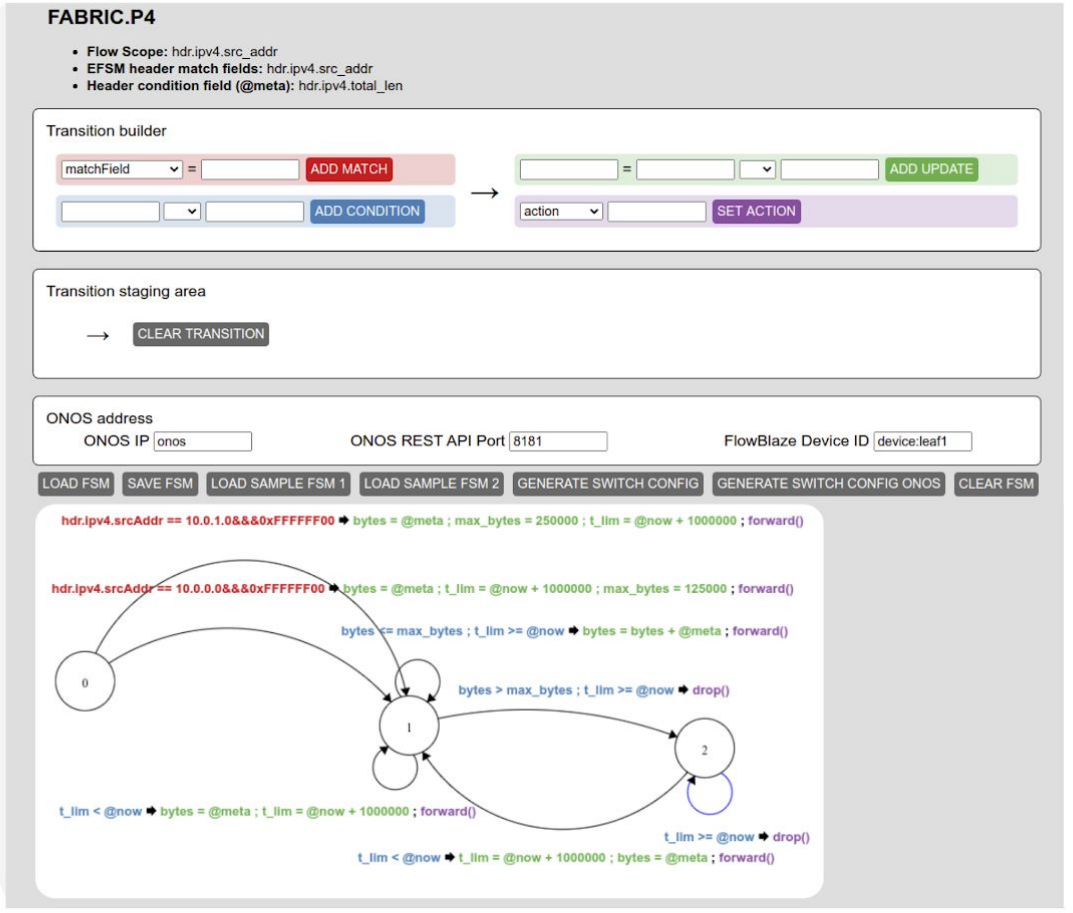
- Add `flowblaze.p4` library to `fabric.p4` pipeline
- Exploit current Trellis apps to program the rest of the fabric.p4 pipeline (routing, bridging, link discovery…)
- FlowBlaze ONOS app to control the FlowBlaze portion of the pipeline





FlowBlaze.p4 + ONOS WorkFlow





FABRIC.P4

- **Flow Scope:** hdr.ipv4.src_addr
- **EFSM header match fields:** hdr.ipv4.src_addr
- **Header condition field (@meta):** hdr.ipv4.total_len

matchField =

action

ONOS address

ONOS IP onos   ONOS REST API Port 8181   FlowBlaze Device ID device:leaf1

```
hdr.ipv4.srcAddr == 10.0.1.0&&&0xFFFFFF00 ➡ bytes = @meta ; max_bytes = 250000 ; t_lim = @now + 1000000 ; forward()
hdr.ipv4.srcAddr == 10.0.0.0&&&0xFFFFFF00 ➡ bytes = @meta ; t_lim = @now + 1000000 ; max_bytes = 125000 ; forward()
bytes <= max_bytes ; t_lim >= @now ➡ bytes = bytes + @meta ; forward()
bytes > max_bytes ; t_lim >= @now ➡ drop()
t_lim < @now ➡ bytes = @meta ; t_lim = @now + 1000000 ; forward()
t_lim >= @now ➡ drop()
t_lim < @now ➡ t_lim = @now + 1000000 ; bytes = @meta ; forward()
```

Daniele Moro
daniele.moro@polimi.it

ANTLAB | ADVANCED NETWORK TECHNOLOGIES LAB

https://github.com/ANTLab-polimi/ONOS-flowblaze

ANTLAB

ADVANCED
NETWORK
TECHNOLOGIES
LAB